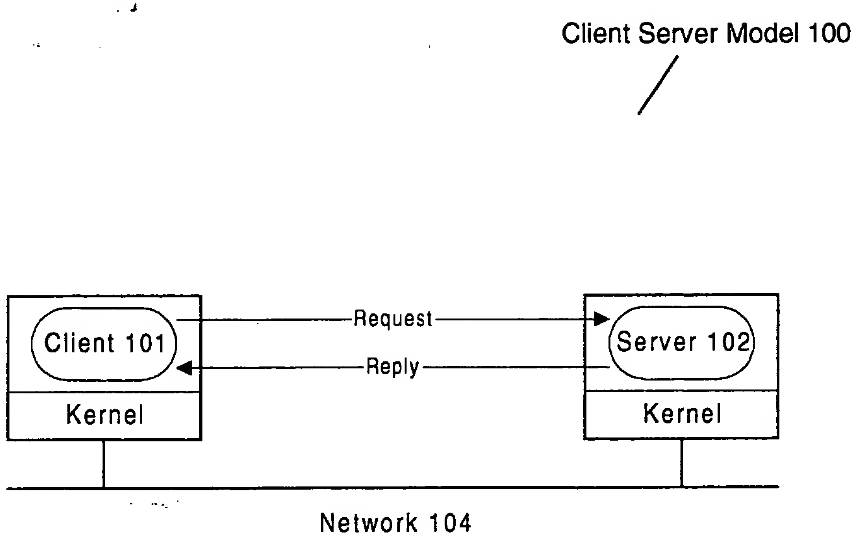


Fig. 1 - The client-server model PRIOR ART

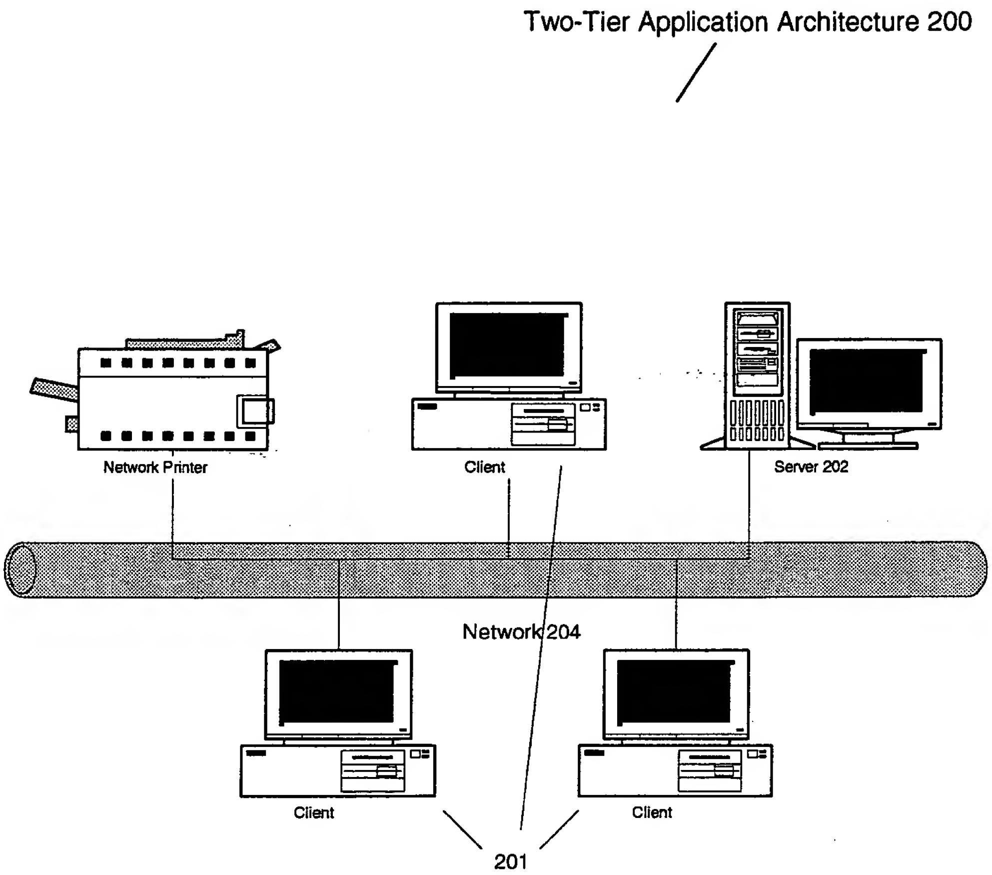


Fig. 2 PRIOR ART

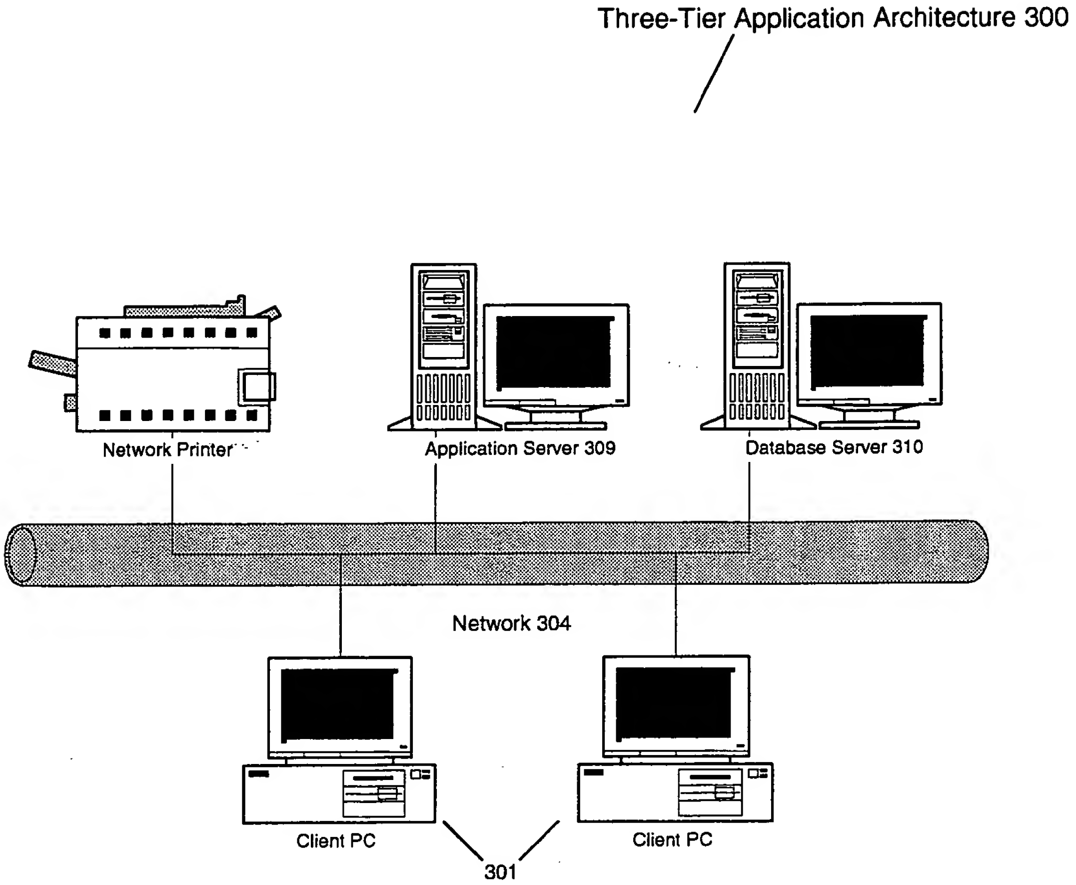
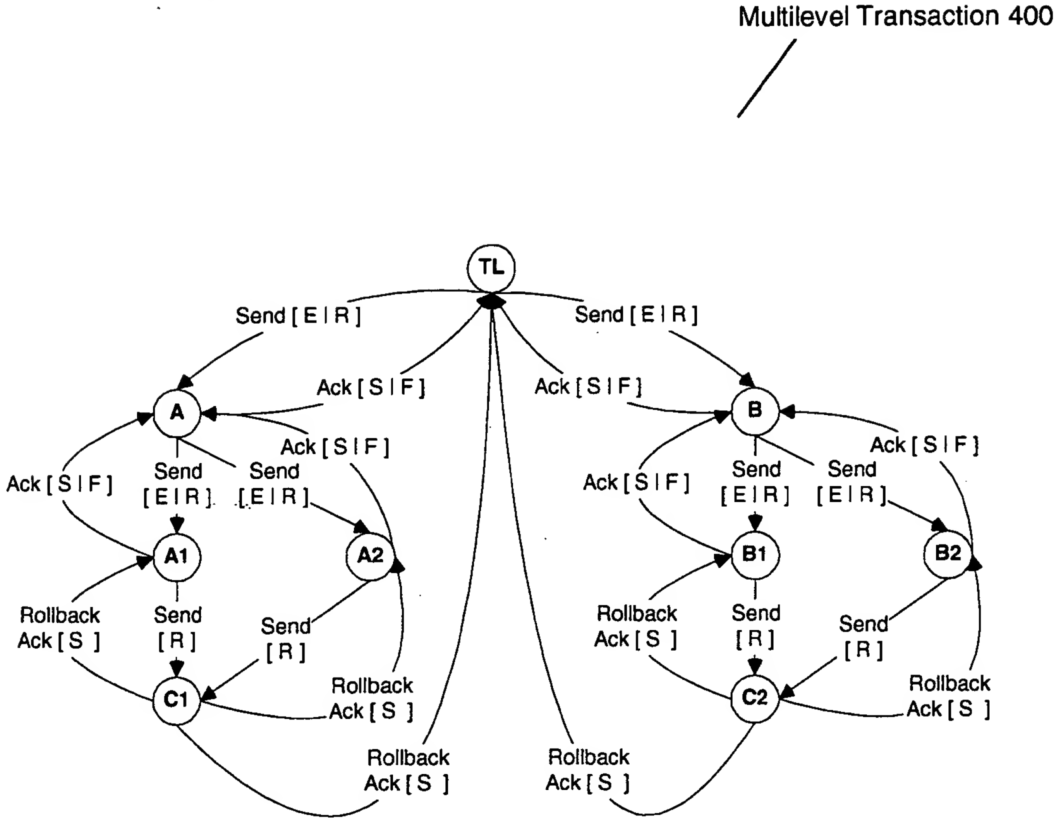


Fig. 3 PRIOR ART



Ack [S I F] = Acknowledgement [Success I Failure]

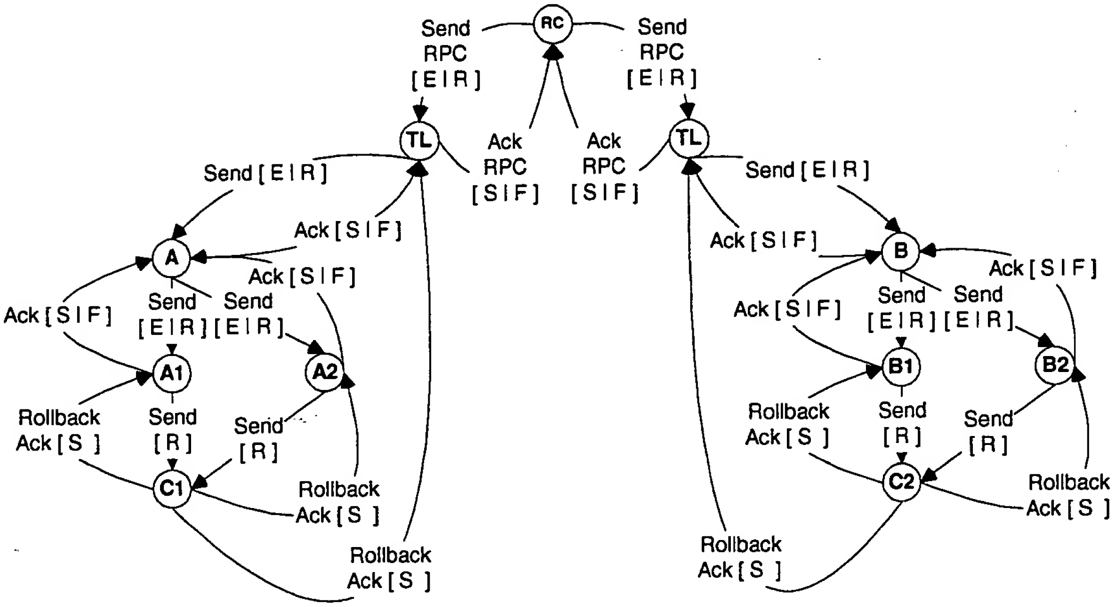
Rollback Ack [S I F] = Rollback Acknowledgement [Success]

Send [E I R] = Send [Execute I Rollback]

Send [R] = Send [Rollback]

Fig. 4

Com d Multi Level Transaction 500



Ack [S I F] = Acknowledgement [Success I Failure]

Rollback Ack [S I F] = Rollback Acknowledgement [Success]

Ack RPC [S I F] = Acknowledgement RPC [Success I Failure]

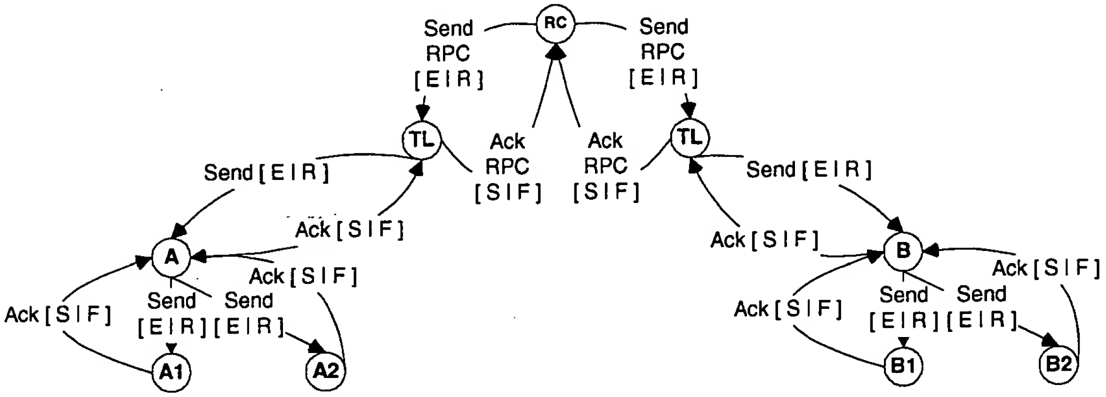
Send [E I R] = Send [Execute I Rollback]

Send [R] = Send [Rollback]

Send RPC [E I R] = Send RPC [Execute I Rollback]

Fig. 5

Compound Nested Transaction 600



Ack [S I F] = Acknowledgement [Success I Failure]

Ack RPC [S I F] = Acknowledgement RPC [Success I Failure]

Send [E I R] = Send [Execute I Rollback]

Send RPC [E I R] = Send RPC [Execute I Rollback]

Fig. 6

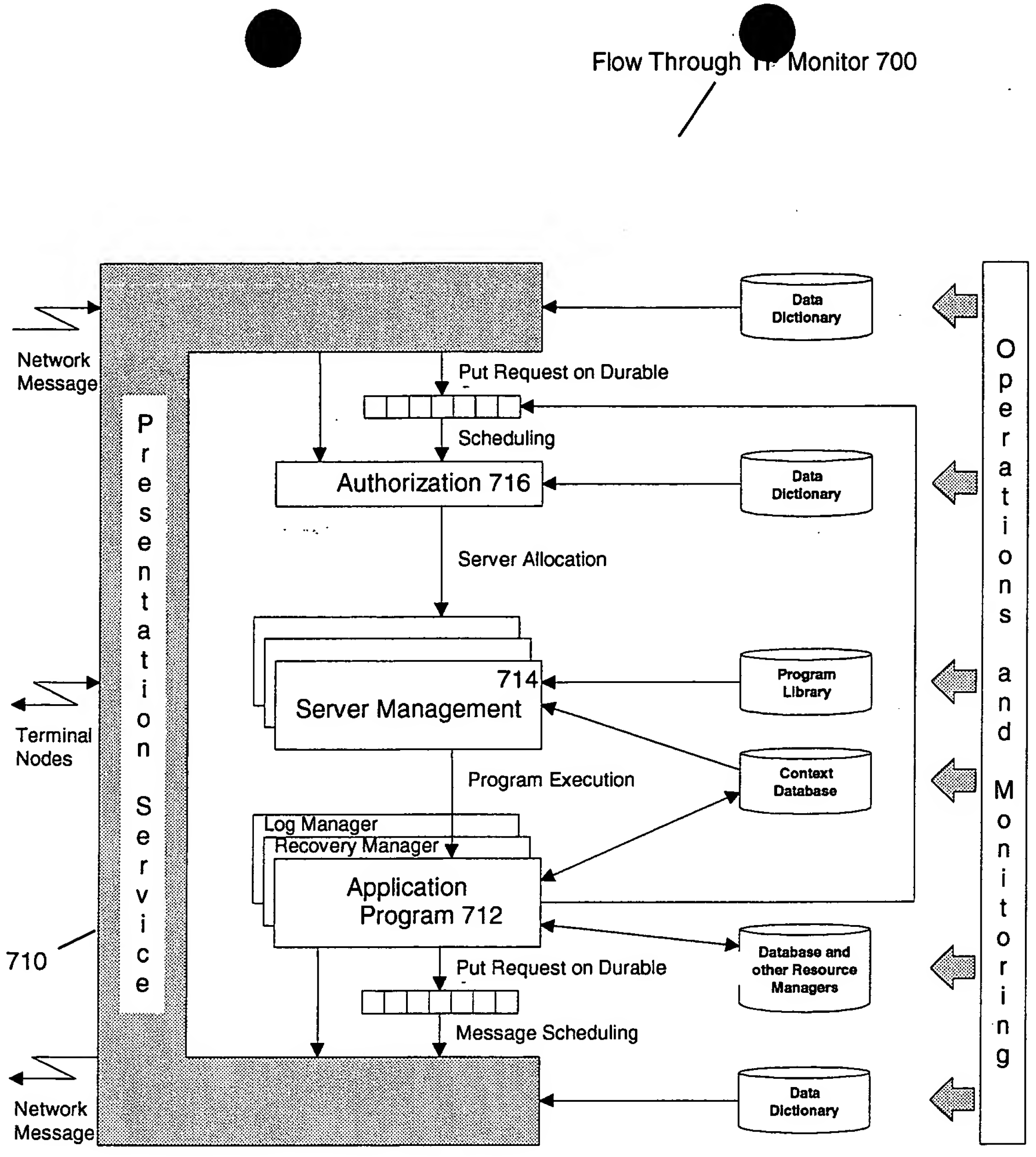


Fig. 7 PRIOR ART

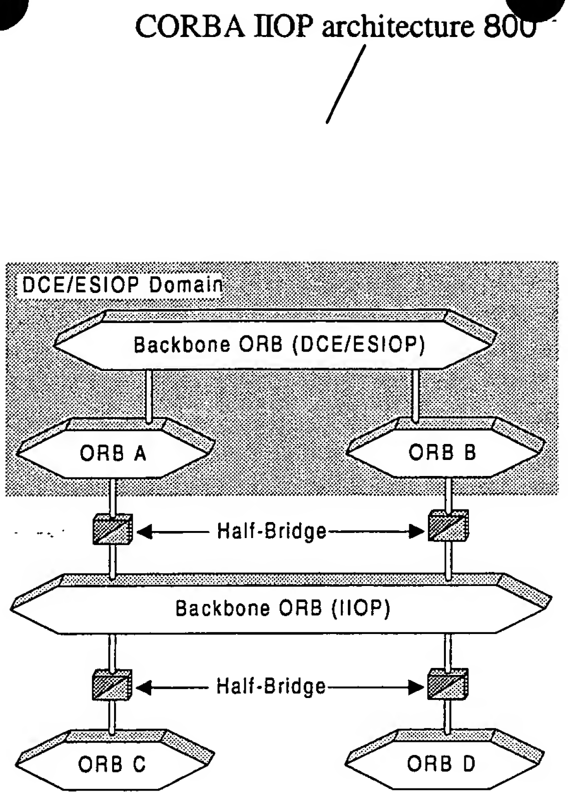


Fig. 8 PRIOR ART

Corba Object Transaction Service 900

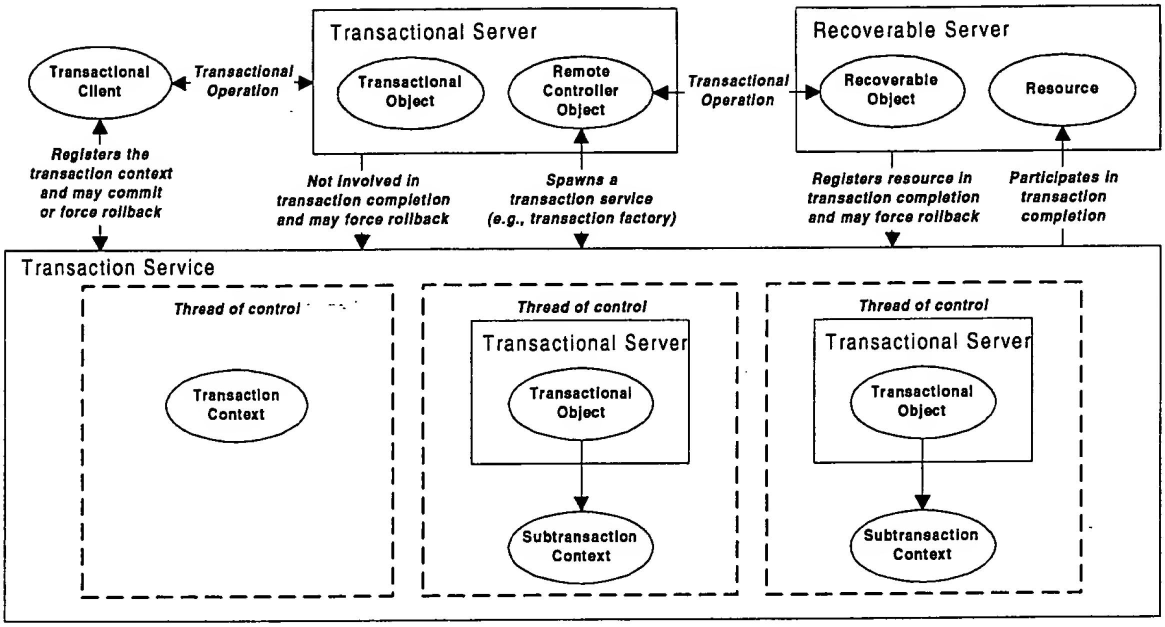


Fig. 9

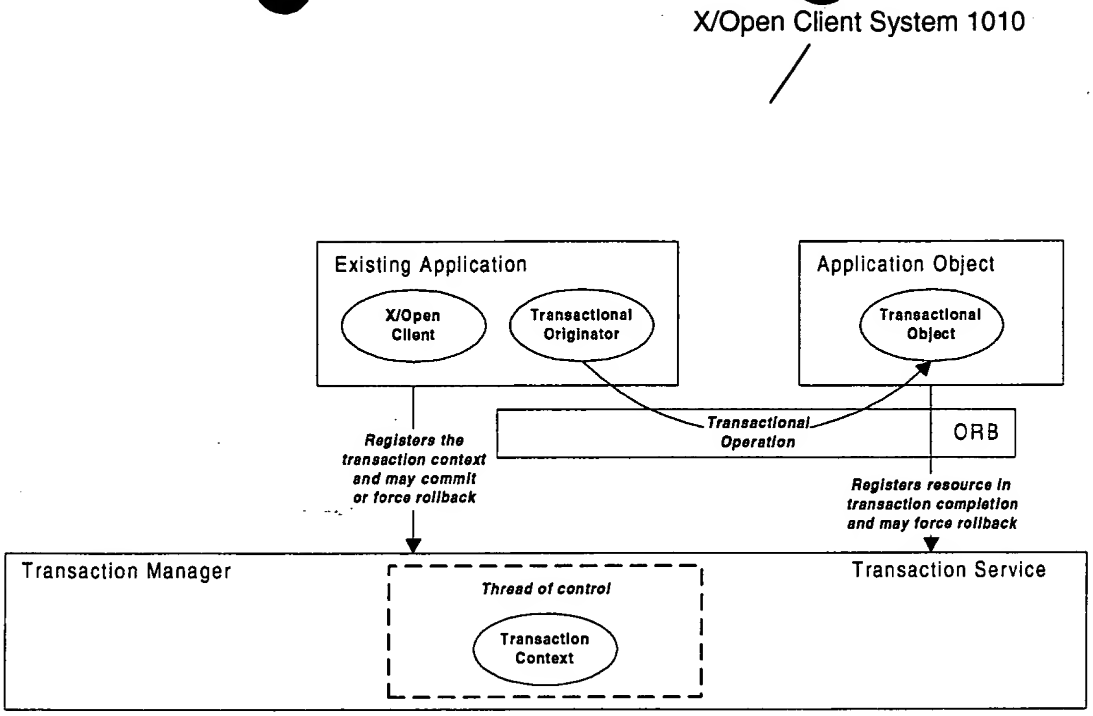


Fig. 10 PRIOR ART

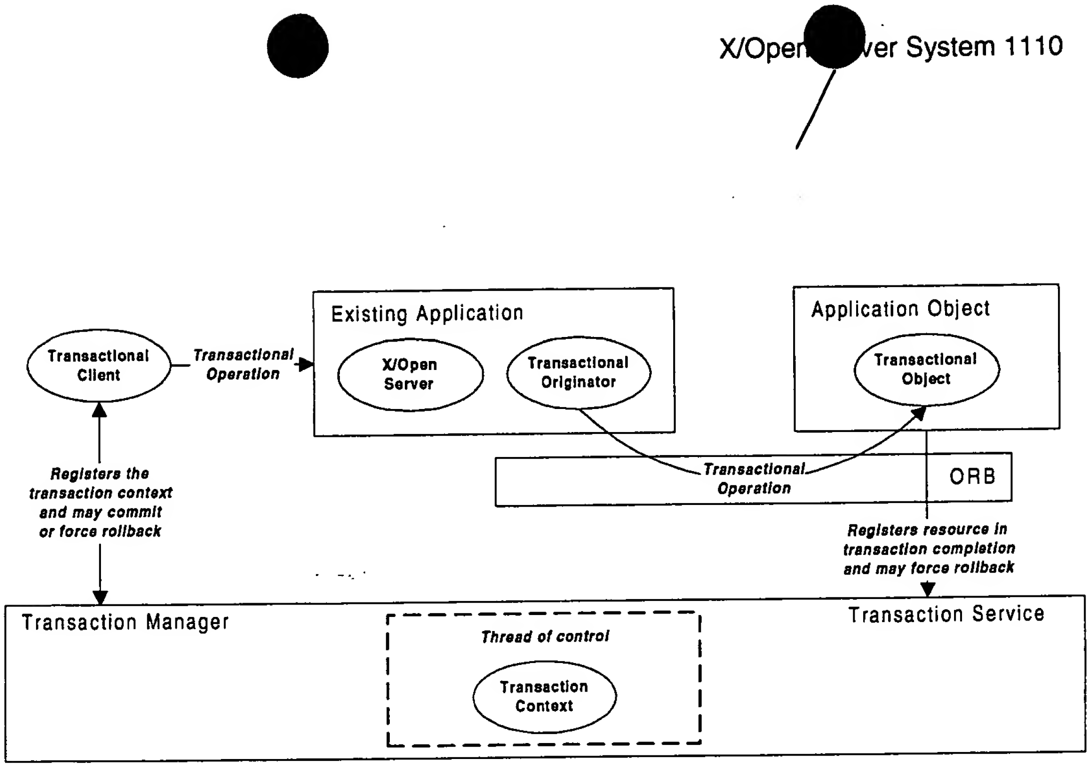


Fig. 11 PRIOR ART

Comparison of Computing Styles 1200

	Batch Processing	Time-Sharing Processing	Realtime Processing	Client-Server	Transaction-Oriented Processing
Data	Private	Private	Private	Shared	Shared
Duration	Long	Long	Very Short	Long	Short
Guarantees of Reliability	Normal	Normal	Very High	Normal	Very High
Guarantees of Consistency	None	None	None	None (?)	ACID
Work Pattern	Regular	Regular	Random	Random	Random
Number of Work Sources or Destinations	10	100	1000	100	10000
Services Provided	Virtual Processor	Virtual Processor	Simple Function	Simple Request	Simple or Complex Request
Performance Criteria	Throughput	Response Time	Response Time	Throughput & Response Time	Throughput & Response Time
Availability	Normal	Normal	High	High	High
Unit of Authorization	Job	User	None(?)	Request	Request

Fig. 12 PRIOR ART

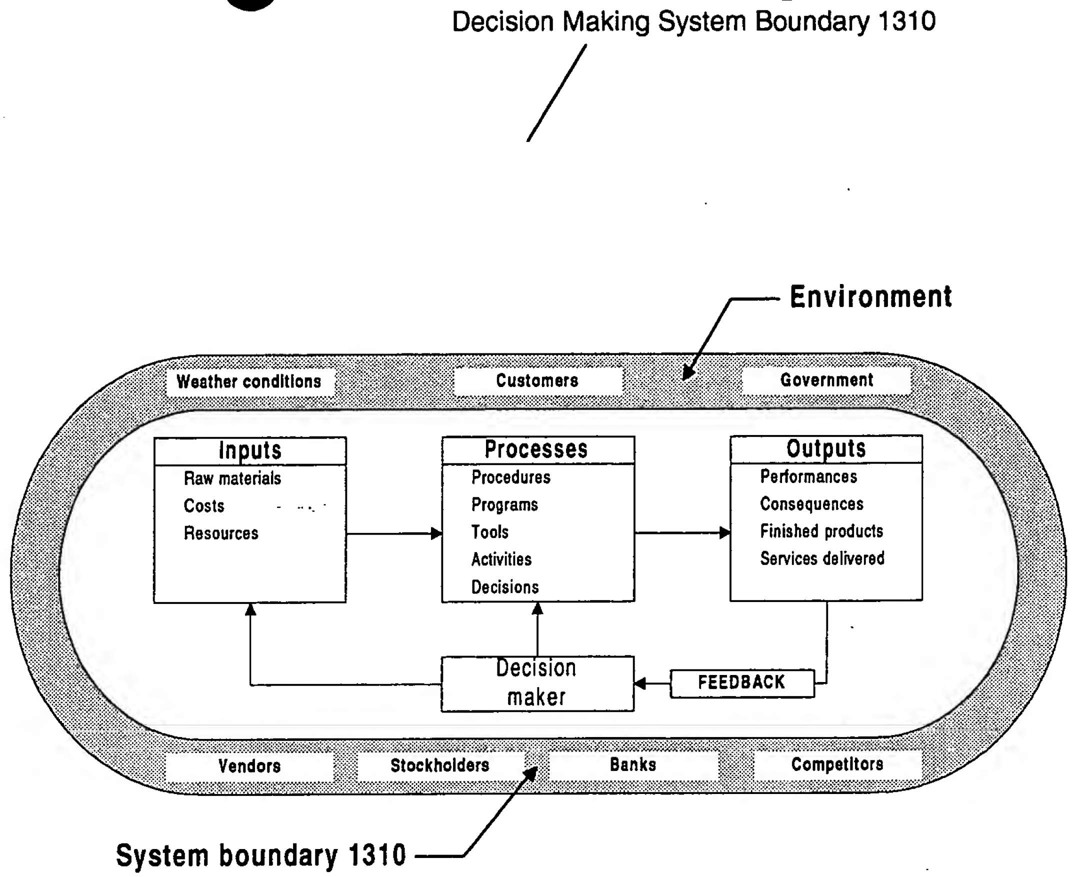


Fig. 13 PRIOR ART

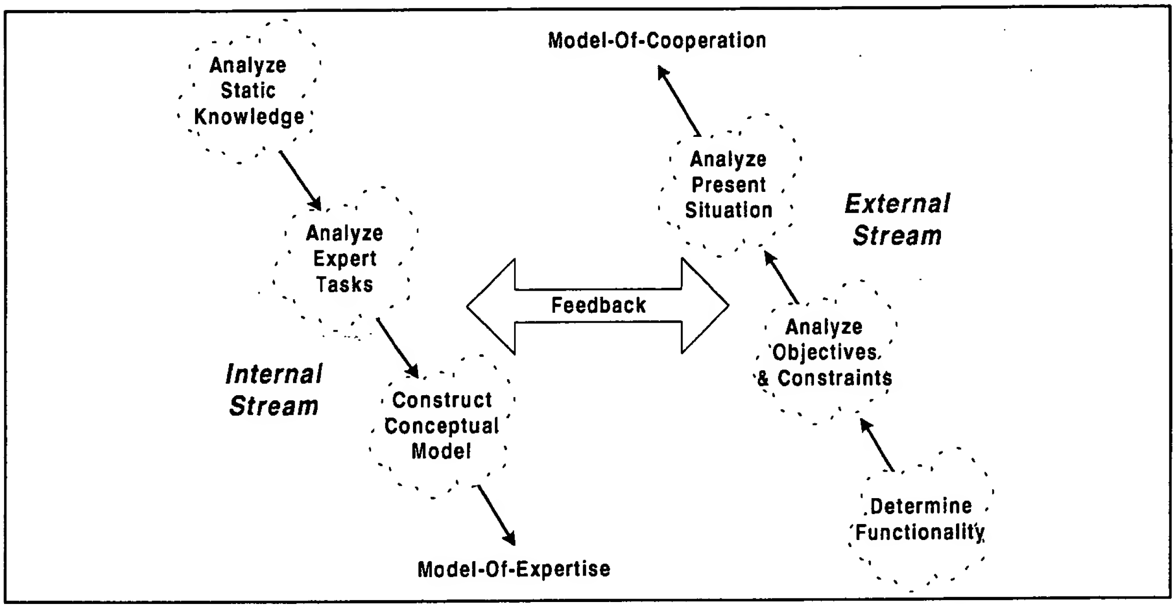


Fig. 14 PRIOR ART

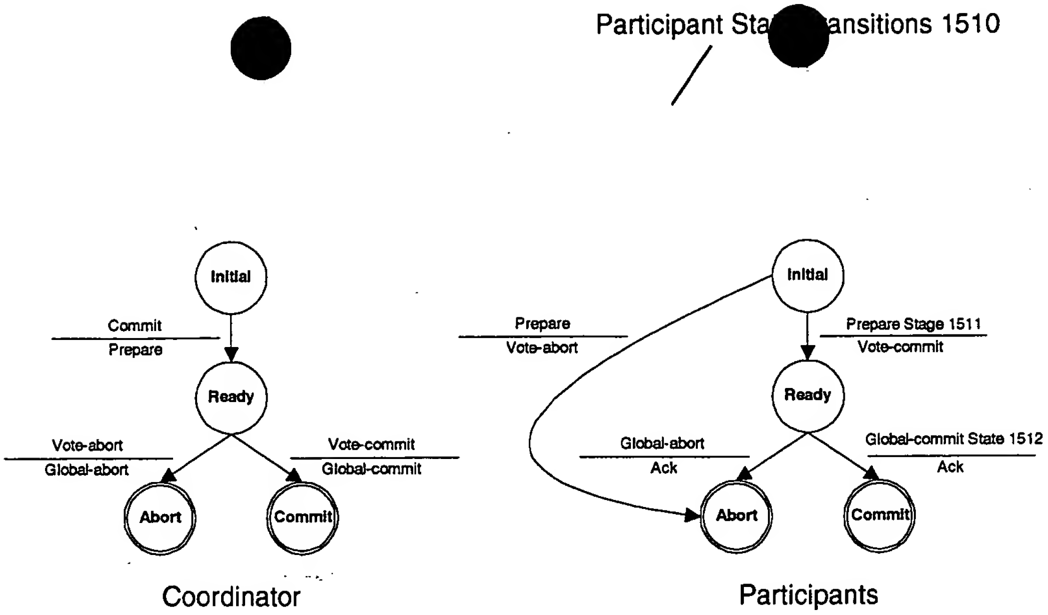


Fig. 15 PRIOR ART

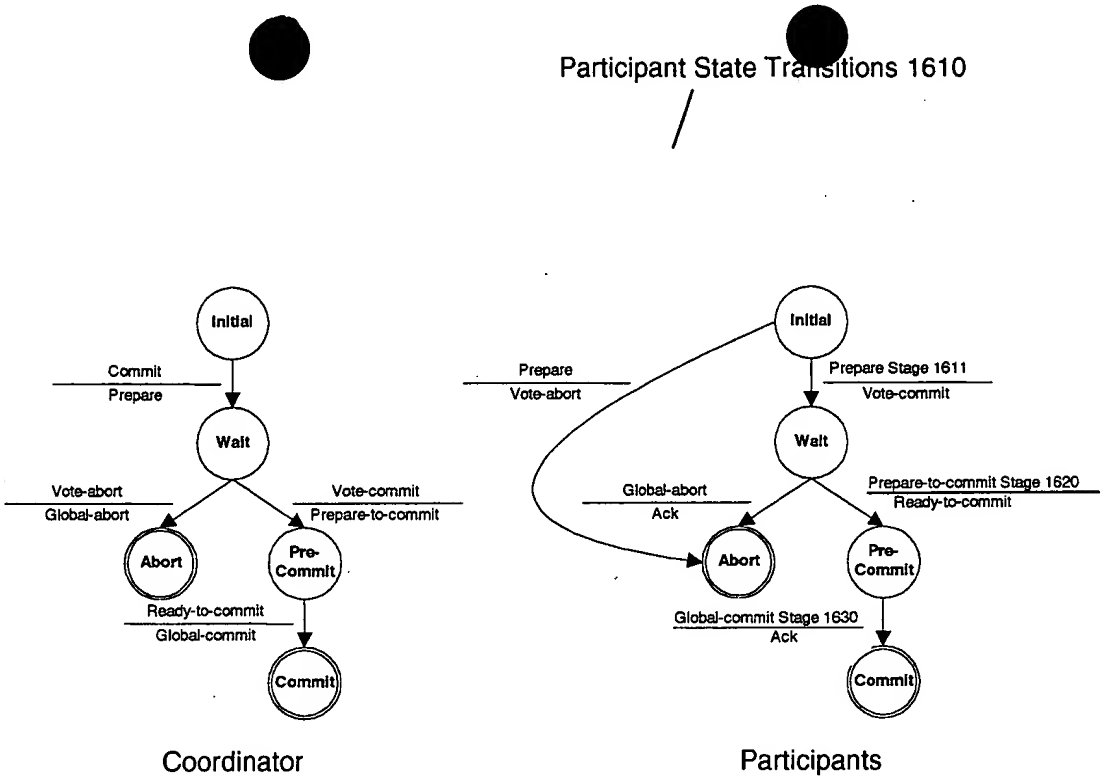


Fig. 16 PRIOR ART

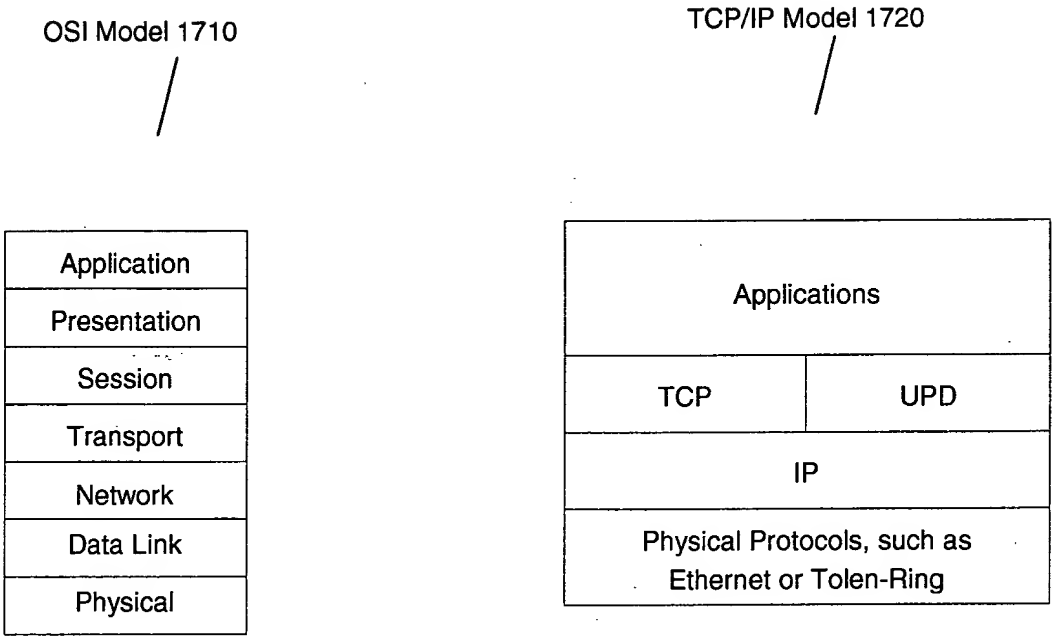


Fig. 17 PRIOR ART

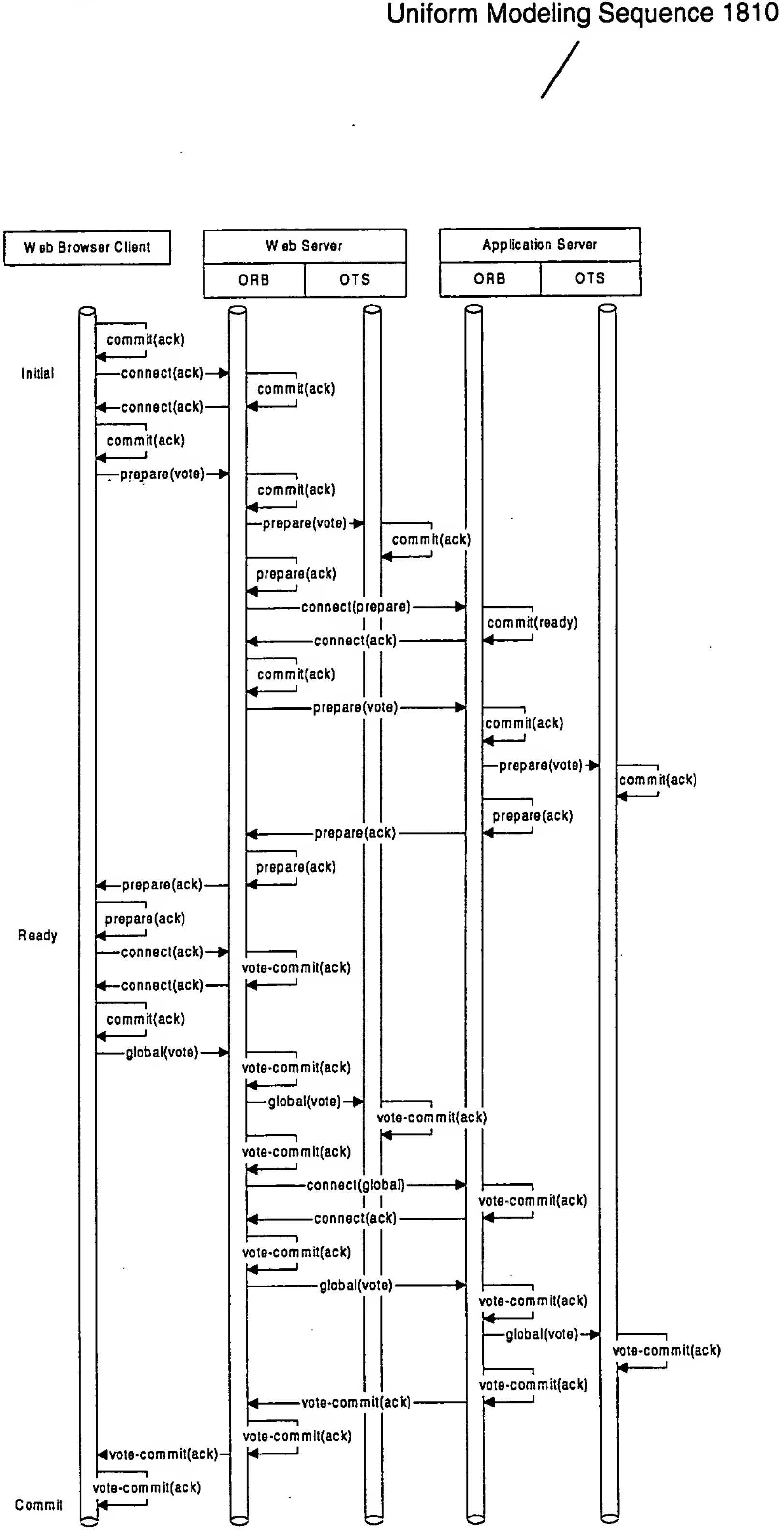
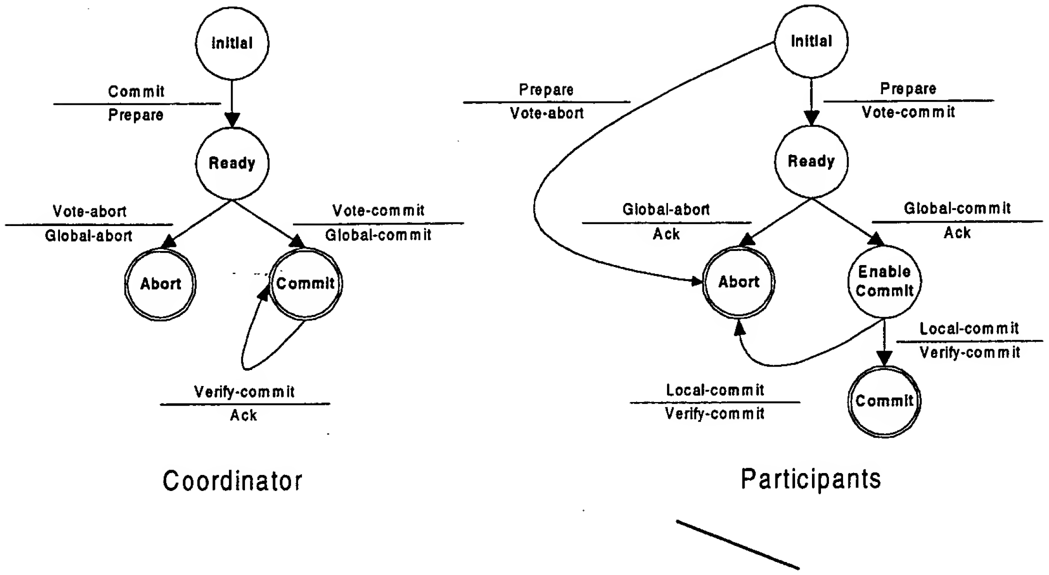

Uniform Modeling Sequence 1810
Web Browser Client
Web Server
ORB
OTS
Application Server
ORB
OTS
Initial
Ready
Commit
commit(ack)
connect(ack)
commit(ack)
connect(ack)
commit(ack)
prepare(vote)
commit(ack)
prepare(vote)
commit(ack)
prepare(ack)
connect(prepare)
commit(ready)
connect(ack)
commit(ack)
prepare(vote)
commit(ack)
prepare(vote)
commit(ack)
prepare(ack)
prepare(ack)
prepare(ack)
prepare(ack)
prepare(ack)
connect(ack)
vote-commit(ack)
connect(ack)
commit(ack)
global(vote)
vote-commit(ack)
global(vote)
vote-commit(ack)
vote-commit(ack)
connect(global)
vote-commit(ack)
connect(ack)
vote-commit(ack)
global(vote)
vote-commit(ack)
global(vote)
vote-commit(ack)
vote-commit(ack)
vote-commit(ack)
vote-commit(ack)
vote-commit(ack)
vote-commit(ack)

Fig. 18A



Orthogonal State Transitions 1850

Fig. 18B

Initial_Transaction Procedure Definition 1910

```
METHOD
------
MEMBER PROCEDURE INITIAL_TRANSACTION
Argument Name                  Type                     In/Out
------------------------------ ----------------------- ------
TRANSACTION_ID                 NUMBER                   IN
TRANSACTION_PARENT_ID          NUMBER                   IN
TRANSACTION_SOURCE             VARCHAR2                 IN
TRANSACTION_DESTINATION        DESTINATION              IN
TRANSACTION_TIME_STAMP         DATE                     IN
TRANSACTION_QUANTUM            NUMBER                   IN
TRANSACTION_TYPE               VARCHAR2                 IN
TRANSACTION_STATUS             VARCHAR2                 IN/OUT
TRANSACTION_NAME               VARCHAR2                 IN
DML_ACTION                     VARCHAR2                 IN
DML_ATTRIBUTES                 ATTRIBUTE                IN
OBJ_NAME                       VARCHAR2                 IN
OBJ_ATTRIBUTES                 ATTRIBUTE                IN
WHERE_CLAUSE                   ATTRIBUTE                IN
```

Fig. 19

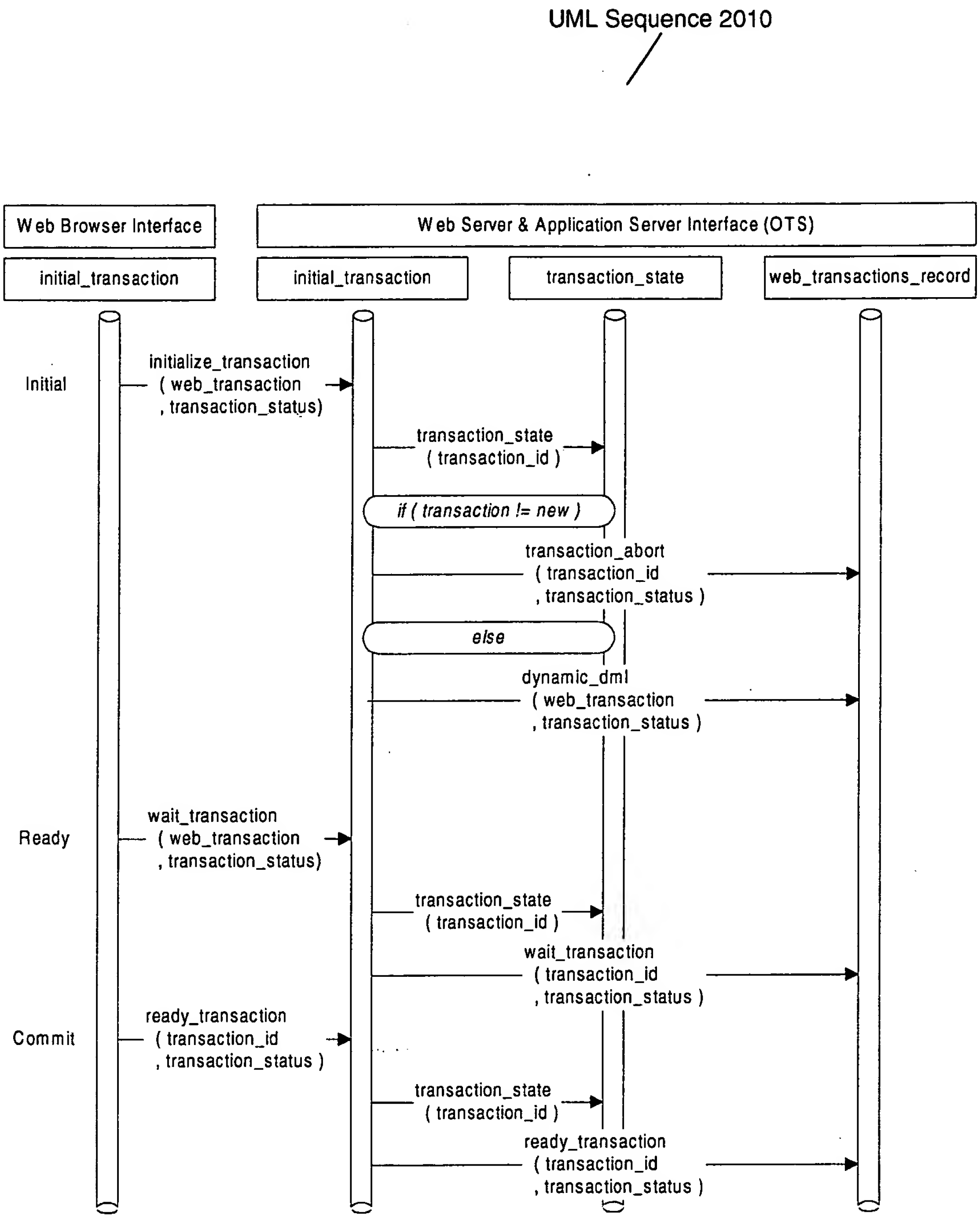

UML Sequence 2010
Web Browser Interface
Web Server & Application Server Interface (OTS)
initial_transaction
initial_transaction
transaction_state
web_transactions_record
Initial
initialize_transaction (web_transaction , transaction_status)
transaction_state (transaction_id)
if (transaction != new)
transaction_abort (transaction_id , transaction_status)
else
dynamic_dml (web_transaction , transaction_status)
Ready
wait_transaction (web_transaction , transaction_status)
transaction_state (transaction_id)
wait_transaction (transaction_id , transaction_status)
Commit
ready_transaction (transaction_id , transaction_status)
transaction_state (transaction_id)
ready_transaction (transaction_id , transaction_status)

Fig. 20

Initial_Transaction Procedure Definition 2110

```
METHOD
------
MEMBER PROCEDURE INITIAL_TRANSACTION
Argument Name                    Type                     In/Out
------------------------------   ----------------------   -----
TRANSACTION_ID                   NUMBER                   IN
TRANSACTION_STATUS               VARCHAR2                 IN/OUT
TRANSACTION_SOURCE               VARCHAR2                 IN
TRANSACTION_DETAIL               TRANSACTION              IN
```

Fig. 21

Transaction Object Definition 2210

```
OBJECT TYPE (TRANSACTION)
-------------------------
Argument Name                   Type                   In/Out
-----------------------------   ----------------------  -----
TRANSACTION_ID                  NUMBER                 IN
TRANSACTION_PARENT_ID           NUMBER                 IN
TRANSACTION_SOURCE              VARCHAR2               IN
TRANSACTION_DESTINATION         DESTINATION            IN
TRANSACTION_TIME_STAMP          DATE                   IN
TRANSACTION_QUANTUM             NUMBER                 IN
TRANSACTION_TYPE                VARCHAR2               IN
TRANSACTION_STATUS              VARCHAR2               IN/OUT
TRANSACTION_NAME                VARCHAR2               IN
DML_ACTION                      VARCHAR2               IN
DML_ATTRIBUTES                  ATTRIBUTE              IN
OBJ_NAME                        VARCHAR2               IN
OBJ_ATTRIBUTES                  ATTRIBUTE              IN
WHERE_CLAUSE                    ATTRIBUTE              IN
```

Fig. 22

CORBA Transaction-Messaging Matrix 2310

Number of Components Involved			Number of Messages			
Web Browser	Web Server	Application Server	Initial	Ready	Commit	Total Messages
1	1	1	23	23	23	69
1	1	2	34	34	34	102
1	1	3	45	45	45	135
1	1	4	56	56	56	168
1	1	5	67	67	67	201

Fig. 23

Expository ACID Compliant Transaction Archetecture 2400

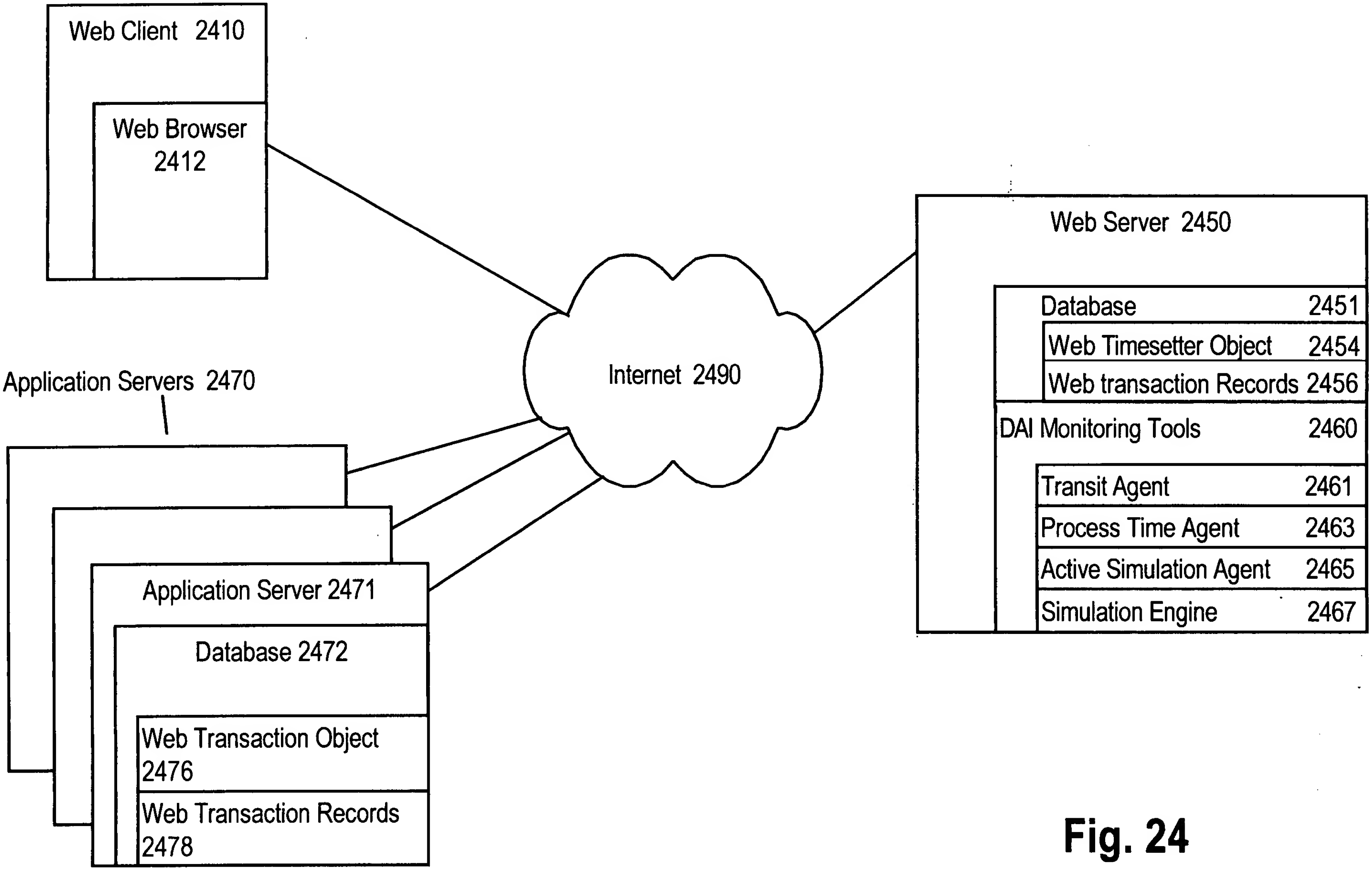
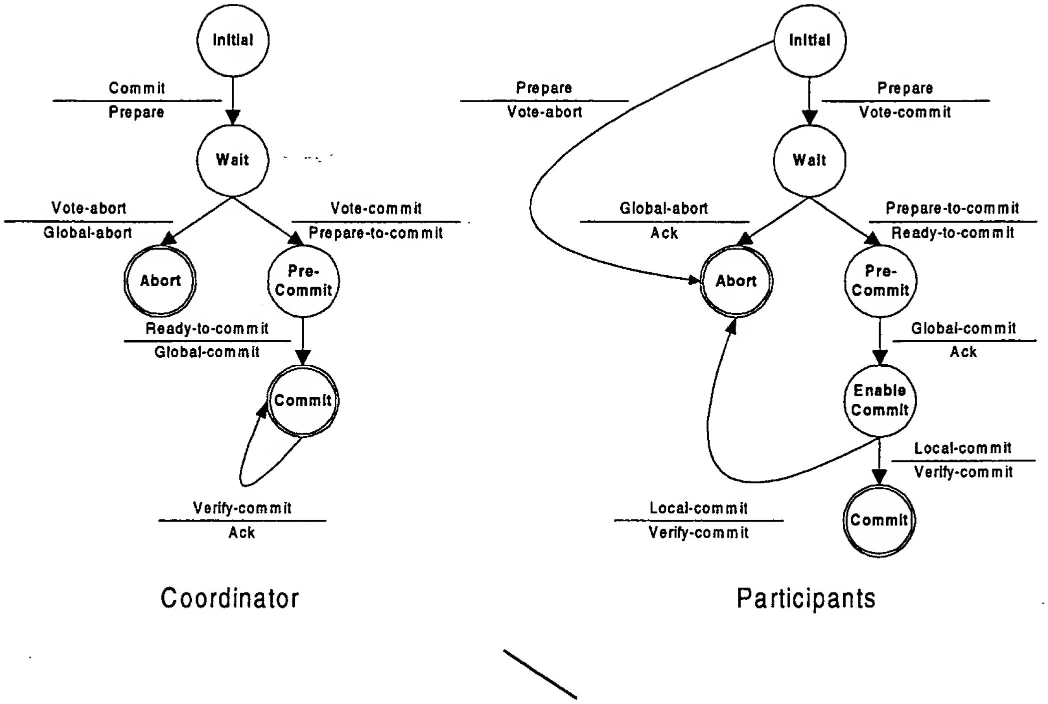


Fig. 24



Coordinator

Participants

Orthogonal State Transitions 2500

Fig. 25A

Fig. 25B

Orthogonal State Transitions 2510

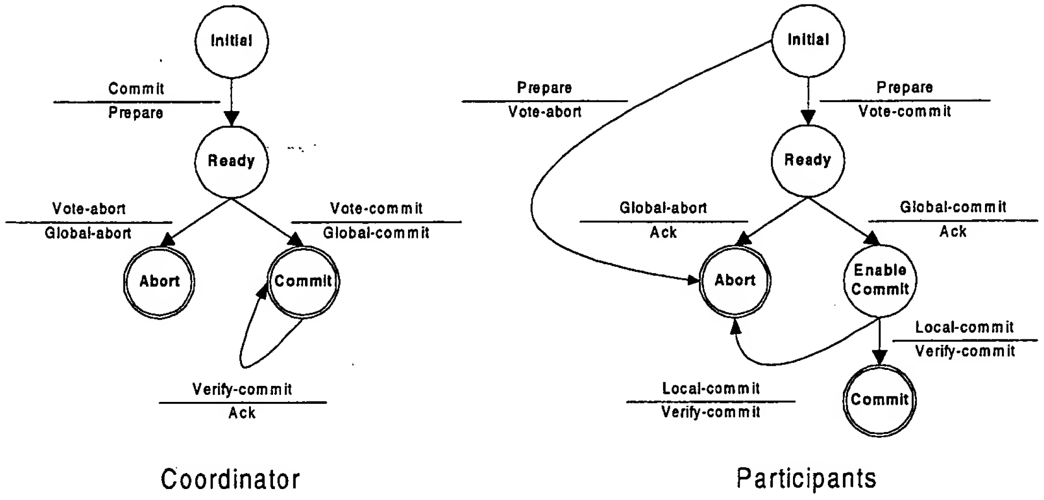


Fig. 25C

Orthogonal State Transitions 2520

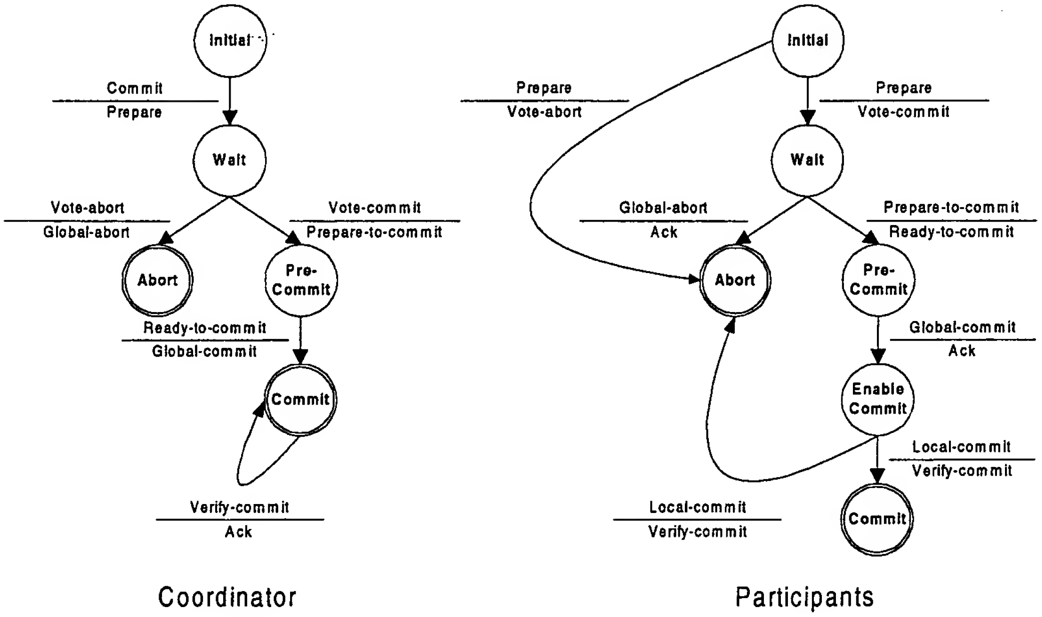


Fig. 26

Asynchronous Transaction Object Management System Architecture Diagram 2600

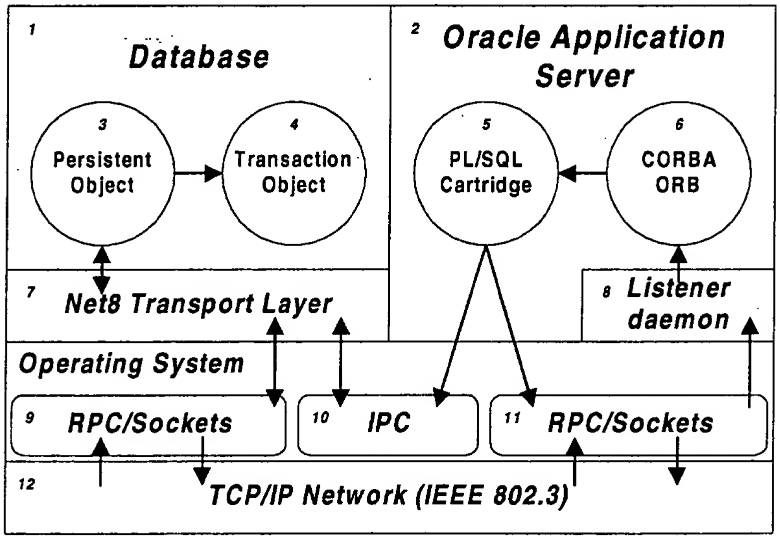


Fig. 27

Asynchronous Transaction Object Management System operating system architecture diagram 2700

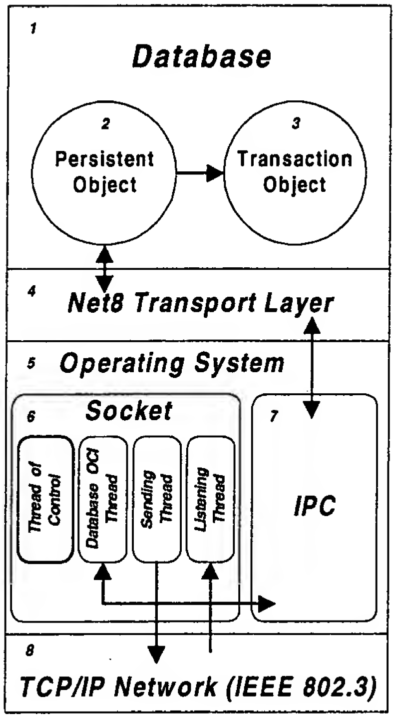
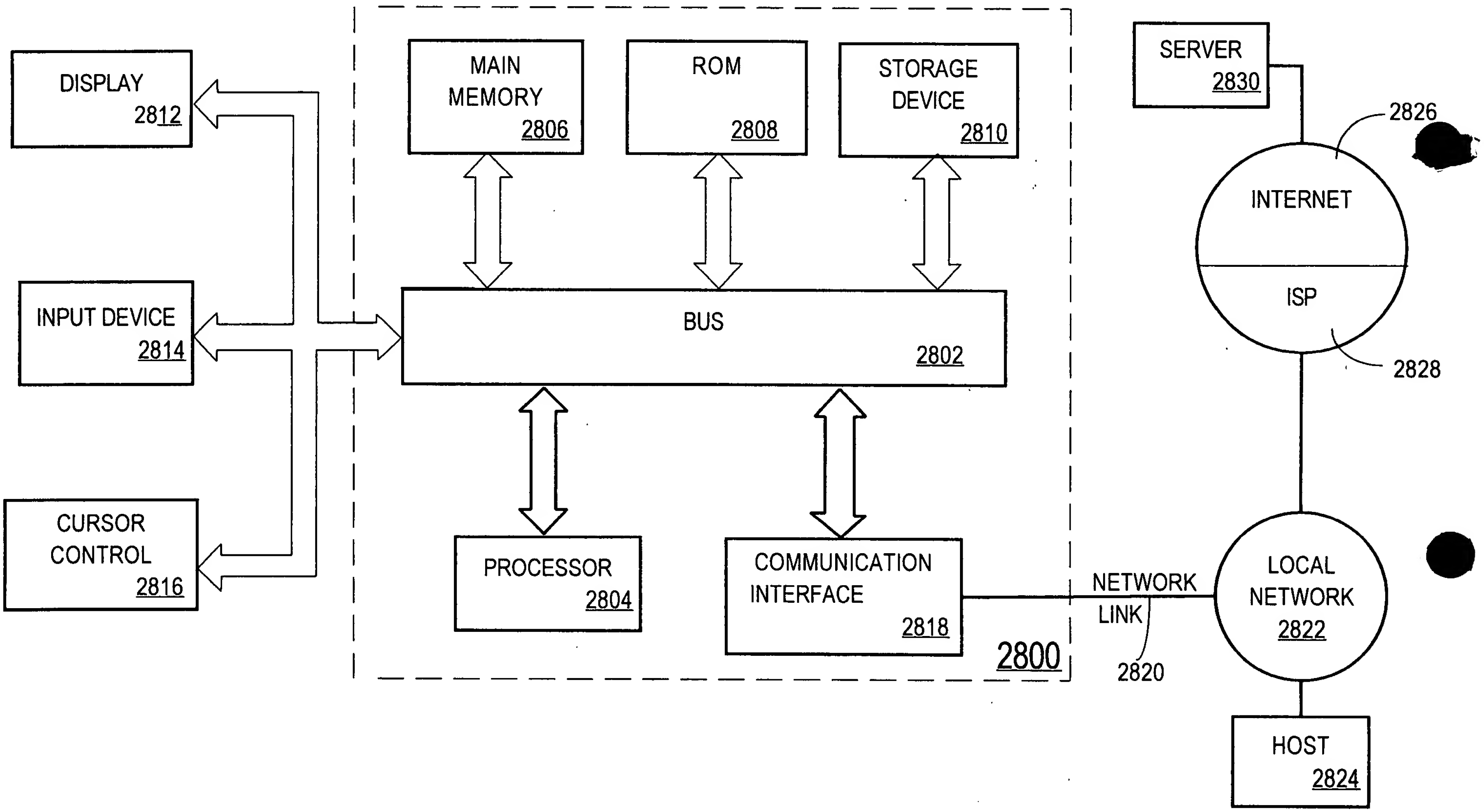


FIG. 28


DISPLAY
2812
INPUT DEVICE
2814
CURSOR
CONTROL
2816
MAIN
MEMORY
2806
ROM
2808
STORAGE
DEVICE
2810
BUS
2802
PROCESSOR
2804
COMMUNICATION
INTERFACE
2818
2800
NETWORK
LINK
2820
SERVER
2830
2826
INTERNET
ISP
2828
LOCAL
NETWORK
2822
HOST
2824